execute a many-rows method to redistribute the one or more rows.

REMARKS

Claims 1-26 are pending.

In the following, the Examiner's comments, when included, are presented in bold, indented type, followed by the Applicants' remarks.

> 1. **Claim 26 is objected to because of the following informalities:**
> **On line 4 of claim 26, "providing access to on or more" should be replaced by -- providing access to one or more--. This appears to be a typographical error.**
> **Appropriate correction is required.**

Claim 26 has been amended as suggested by the Examiner.

> 1. **Claim 26 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.**
> **Claim 26 recites "a set of one or more database tables residing on the one or more nodes, the one or more database tables containing information organized by geographic location". Although paragraph [0016] of the specification of the instant invention describes tables stored across multiple data-storage facilities, it does not describe that the tables contain information organized by geographic location. Therefore, claim 26 contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.**

Claim 26 has been amended to remove the following language: "the one or more database tables containing information organized by geographic location." Applicant considers this a broadening amendment. Withdrawal of this rejection is requested.

> 2. **Claims 18, 25, and 26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.**

Claim 18 recites a "computer executable code for instructing a computer to" perform steps of the method of the instant invention. The recited language "code for" makes it unclear as to whether the steps that the claimed computer executable code is for are limitations in claim 26, or are simply optional. See MPEP § 2111.04.

For purposes of examination, the recited "computer executable code" is interpreted as any code, instructions, program, or software executable by a computer.

Applicant respectfully disagrees and believes that it is clear that the executable code is for the limitations in claim 18. Applicant respectfully requests withdrawal of this rejection.

Claim 25 recites "a memory operable to store a program". The recited language "operable to" makes it unclear as to whether the steps the memory is operable to perform are limitations in the claim, or are simply optional. See MPEP § 2111.04.

For purposes of examination, "a memory operable to store a program" is interpreted as any type of memory which is capable of having stored therein code, instructions, a program, or software.

Claim 25 recites "at least one processor operable to determine". The recited language "operable to" makes it unclear as to whether the steps the processor is operable to perform are limitations in the claim, or are simply optional. See MPEP § 2111.04.

For purposes of examination, "at least one processor operable to determine" is interpreted as any processor capable of executing instructions.

Claim 25 recites "the at least one processor capable of". The recited language "capable of" makes it unclear as to whether the steps the processor is capable of performing are limitations in the claim, or are simply optional. See MPEP § 2111.04.

For purposes of examination, "the at least one processor capable of" is interpreted as any processor capable of executing instructions.

Claim 25 has been amended to address these comments. Claim 25 has also been amended to correct a typographical error ("singe" to "single"), to eliminate the word "also," and to add the word "otherwise." Applicant respectfully requests withdrawal of these rejections.

Claim 26 recites a "plurality of virtual processes operable to" perform steps of the method of the present invention. The recited language "operable to" makes it unclear as to whether the steps that the claimed virtual processes are operable to perform are limitations in claim 26, or are simply optional. See MPEP § 2111.04.

For purposes of examination, "plurality of virtual processes operable to" is interpreted as any code, instructions, program, or software executing on a computer.

Claim 26 has been amended to address this comment. Applicant respectfully requests withdrawal of this rejection.

Claim 25 is rejected under 35 U.S.C. 102(b) as being anticipated by Chen et al (U.S. Patent Number 5,819,083).

As to claim 25, Chen et al teaches a relational database management system (see Abstract), comprising:

a memory operable to store a program accessible to one or more of a plurality of transmitting modules, the program capable of managing a redistribution of one or more rows associated with one or more database tables (item 25 in Figure 2, showing the memory of the computer system in Figure 2); and

at least one processor operable to determine the number of transmitting modules on which the program was invoked, the at least one processor capable of executing a few-rows row redistribution method to redistribute the one or more rows if the program was invoked on a singe transmitting module, the at least one processor also capable of executing a many-rows row redistribution method to redistribute the one or more rows (item 22 in Figure 2, showing the processor of the computer system in Figure 2).

Applicant respectfully requests that this rejection be withdrawn in light of the amendments to claim 25.

Claims 1-24 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al (U.S. Patent Number 5,819,083) and further in view of Shatdal et al (Shatdal, A., Naughton, J. F., "Adaptive Parallel Aggregation Algorithms", 1995, SIGMOD '95, pp. 104-114).

As to claim 1, Chen et al teaches a method for redistributing data in a relational data base management system (see Abstract), comprising:

allocating a buffer associated with a transmitting processing module, the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables (col. 2 lines 47-50, allocating a communication buffer for transferring rows between nodes, enabling redistribution of rows);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows:

storing one or more rows of a database table in the allocated buffer (col. 5 lines 34-37, indicating that portions of database tables are stored in the buffer);

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (col. 6 lines 48-51, sending row data contained in a buffer to a node);

Chen et al does not explicitly teach comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows;

executing a many-rows method to redistribute the one or more rows.

Shatdal et al teaches comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (p. 106 col. 1 lines 12-16, determining if a table will fit in allocated memory);

executing a many-rows method to redistribute the one or more rows (p. 106 col. 1 lines 12-16, the allocated memory is not large enough to fit the table, tuples (i.e. rows) are partitioned into multiple buckets. This is a many-rows method of redistributing, since it is executed when there the number of rows is too large to fit into allocated memory.).

Therefore, it would have been obvious to one skilled in the art at the time the invention was made to have modified the method of redistributing rows in a database taught by Chen et al by the step of comparing the size of a buffer to the size of a table of rows to be stored in the buffer, because comparing the size of a buffer to the size of a table of rows to be stored in the buffer white redistributing rows in a database enables dynamic adaptation to grouping selectivities at query evaluation time (Shatdal et al p. 104 col. 1 lines 7-13).

Applicant respectfully disagrees. Chen does not teach or suggest storing one or more rows of the allocated buffer if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows, as required by claim 1. Instead, according to the portion of Chen cited in the Office Action, Chen appears to always (i.e., not conditioned on the size of the buffer or the portion of the buffer to be occupied) store portions of tables prior to transmission.

Further, Shatdal does not teach or suggest executing a many-rows method to redistribute the one or more rows, as required by claim 1. A many-rows redistribution method is not simply a method that is executed when the number of rows is too large to fit into allocated memory, as suggested in the Office Action. A description of an example of a many-rows method to distribute the one or more rows is in Fig. 5 and accompanying text of the specification. The portion of Shatdal cited in the Office Action does not teach or suggest such a method. It is

admitted in the Office Action that Chen does not disclose a many-rows method to distribute the one or more rows.

Thus, the combination of Chen and Shatdal suggested in the Office Action is missing at least two elements required by claim 1. Consequently, the Office Action fails to make a prima facie case of obviousness. Applicant respectfully requests that this rejection be withdrawn.

As to claims 2 and 12, Chen et al, as modified by Shatdal et al, teaches the message comprises all of the one or more rows stored in the allocated buffer (Chen et al col. 6 lines 48-49).

Claim 2 depends from claim 1 and is patentable for at least the reasons described above for claim 1. Claim 12 depends from claim 10 and is patentable for at least the reasons described below for claim 10. Applicant respectfully requests that this rejection be withdrawn.

As to claims 3, 13, and 19, Chen et al, as modified by Shatdal et al, teaches the transmitting processing module comprises one of a plurality of processing modules associated with a relational database system (Chen et al col. 5 lines 34-37, indicating each node has a transportation layer containing a communication buffer).

Claim 3 depends from claim 1 and is patentable for at least the same reasons described above for claim 1. Claim 13 depends from claim 10 and is patentable for at least the same reasons described below for claim 10. Claim 19 depends from claim 18 and is patentable for at least the reasons described below for claim 18. Applicant respectfully requests that this rejection be withdrawn.

As to claims 4, 14, and 20, Chen et al, as modified by Shatdal et al, teaches the message is communicated to each of a plurality of destination processing elements (Chen et al col. 6 lines 59-60).

Claim 4 depends from claim 1 and is patentable for at least the reasons described above for claim 1. Claim 14 depends from claim 10 and is patentable for at least the reasons described below for claim 10. Claim 20 depends from claim 18 and is patentable for at least the reasons described below for claim 18.

As to claims 5 and 21, Chen et al, as modified by Shatdal et al, teaches the allocated buffer is capable of storing no more than ten (10) rows (Chen et al col. 8 lines 50-54, since the parameters of the expression can be varied to produce an arbitrary value, and since the expression determines the size of the buffer, the invention is capable of allocating a buffer capable of storing no more than ten (10) rows.).

Claim 5 depends from claim 1 and is patentable for at least the reasons described above for claim 1. Claim 21 depends from claim 18 and is patentable for at least the reasons described below for claim 18.

As to claims 6, 16, and 22, Chen et al, as modified by Shatdal et al, teaches the many-rows row redistribution method comprises:

communicating from one or more transmitting modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules (Chen et al col. 5 lines 34-37);

communicating from one or more of the processing modules a ready-to-receive signal to the one or more transmitting modules (Chen et al item 152 in Figure 6, col. 6 lines 10-13);

communicating from the one or more transmitting modules a second signal comprising the one or more rows associated with the database table (Chen et al col. 6 lines 48-51);

after communication of the last row associated with the database table, communicating from the one or more transmitting modules an end-of-data signal to each of the plurality of processing modules (Chen et al col. 6 lines 53-55 and 59-60).

Applicant respectfully disagrees. Chen at col. 5, lines 34-37 discloses establishing a logical link by the transport layers located at each of the nodes but does teach or suggest communicating from one or more transmitting processing modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules, as required by claims 6, 16, and 22. Chen's "establishing a logical link" does not teach or suggest a transmitting processing module, a first signal from the transmitting processing module, or that any such signal is operable to initiate a row receiver task on each of the processing modules.

Chen at col. 6, lines 10-13 discloses a "registration process." This is not the same as communicating from one or more of the processing modules a ready-to-receive signal to the one or more transmitting modules, as required by claims 6, 16, and 22. There is no hint in Chen's "registration process" that the registering node sends a ready-to-receive signal to one or more transmitting modules.

Thus, the combination of Chen and Shatdal suggested in the Office Action is missing at least two elements required by claims 6, 16, and 22. Consequently, the Office Action fails to make a prima facie case of obviousness. Applicant respectfully requests that this rejection be withdrawn.

As to claims 7 and 23, Chen et al, as modified by Shatdal et al, teaches invoking the program on a single transmitting processing module (Chen et al col. 6 lines 46-47, initiating the program).

Claim 7 depends from claim 1 and is patentable for at least the reasons described above for claim 1. Claim 23 depends from claim 18 and is patentable for at least the reasons described below for claim 18. Applicant respectfully requests that this rejection be withdrawn.

As to claim 8, Chen et al, as modified by Shatdal et al, teaches receiving at each of a plurality of destination processing elements a substantially similar set of the one or more rows stored in the allocated buffer (col. 6 lines 48-51, where the rows are substantially similar in that they are contained in the same buffer).

Applicant respectfully disagrees. Claim 8 requires receiving at each of a plurality of destination processing elements a substantially similar set of the one or more rows stored in the allocated buffer. Chen at col. 6, lines 48-51 describes only a single receiving node ("the communication buffer which has been filled by the table scan process is sent to the new node"). Further, claim 8 depends from claim 1 and is patentable for the reasons described above for claim 1. Applicant respectfully requests that this rejection be withdrawn.

As to claims 9, 15, and 24, Chen et al, as modified by Shatdal et al, teaches determining a number of rows to store in the allocated buffer (Chen et al col. 8 lines 51-54, determining the size of the buffer inherently determines the number of rows that can be stored in the buffer).

Applicant respectfully disagrees. Chen at col. 8, lines 51-54 describes determining the minimum sufficient buffer space for a logical link. That does not mean that Chen determines the number of rows that can be stored in the buffer because, for example, the size of rows may not always be the same. Further, the portions of Chen cited in the Office Action do not teach or suggest determining the number or rows to store in the allocated buffer, as required by claims 9, 15, and 24, which is different from the number of rows that can be stored in the buffer.

Further, claim 9 depends from claim 1 and is patentable for the reasons described above for claim 1. Claim 15 depends from claim 10 and is patentable for the reasons described above for claim 10. Claim 24 depends from claim 18 and is patentable for the reasons described above for claim 18. Applicant respectfully requests that this rejection be withdrawn.

As to claim 10, Chen et al teaches a method for redistributing data in a relational data base management system (see Abstract), comprising:

invoking a program on one or more of a plurality of transmitting modules, the program capable of managing a redistribution of one or more rows associated with one or more database tables (col. 2 lines 47-50, redistributing rows on a plurality of nodes);

Chen et al does not explicitly teach if the program was invoked on a single transmitting module:

executing a few-rows redistribution method to redistribute the one or more rows;

otherwise:

executing a many-rows redistribution method to redistribute the one or more rows.

Shatdal et al teaches if the program was invoked on a single transmitting module:

executing a few-rows redistribution method to redistribute the one or more rows (p. 106 col. 1 lines 12-16, if it was not determined that the table is too large to fit in allocated memory, no additional partitioning is done. This is a few-rows method, since it is executed when the number of rows is small enough to fit into allocated memory.);

otherwise:

executing a many-rows redistribution method to redistribute the one or more rows (p. 106 col. 1 lines 12-16, the allocated memory is not large enough to fit the table, tuples (i.e. rows) are partitioned into multiple buckets. This is a many-rows method of redistributing, since it is executed when there the number of rows is too large to fit into allocated memory.).

Therefore, it would have been obvious to one skilled in the art at the time the invention was made to have modified the method of redistributing rows in a database taught by Chen et al by the step of comparing the size of a buffer to the size of a table of rows to be stored in the buffer, because comparing the size of a buffer to the size of a table of rows to be stored in the buffer while redistributing rows in a database enables dynamic adaptation to grouping selectivities at query evaluation time (Shatdal et al p. 104 col. 1 lines 7-13).

Applicant respectfully disagrees. Shatdal does not teach or suggest executing a many-rows method to redistribute the one or more rows, as required by claim 10. A many-rows redistribution method is not simply a method that is executed when there the number of rows is too large to fit into allocated memory, as suggested in the Office Action. A description of an example of a many-rows method to distribute the one or more rows is in Fig. 5 and accompanying text of the specification. The portion of Shatdal cited in the Office Action does not teach or suggest such a method. It is admitted in the Office Action that Chen does not disclose a many-rows method to distribute the one or more rows.

Further, Shatdal does not teach or suggest executing a few-rows method to redistribute the one or more rows, as required by claim 10. A few-rows redistribution method is not simply a method that is executed when the number of rows is small enough to fit into allocated memory, as suggested in the Office Action. A description of an example of a few-rows method to distribute the one or more rows is in Fig. 6 and accompanying text of the specification. The portion of Shatdal cited in the Office Action does not teach or suggest such a method. It is admitted in the Office Action that Chen does not disclose a few-rows method to distribute the one or more rows.

Thus, the combination of Chen and Shatdal suggested in the Office Action is missing at least two elements required by claim 10. Consequently, the Office Action fails to make a prima facie case of obviousness. Applicant respectfully requests that this rejection be withdrawn.

> **As to claim 11, Chen et al, as modified by Shatdal et al, teaches the few-rows row redistribution method comprises:**
>
> **allocating a buffer associated with a transmitting processing module, the transmitting processing module having access to the program, the program associated with a single transmitting module (Chen et al col. 2 lines 47-50, allocating a communication buffer for transferring rows between nodes, enabling redistribution of rows);**
>
> **comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Shatdal et al p. 106 col. 1 lines 12-16, determining if a table will fit in allocated memory);**
>
> **if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows:**
>
> **storing one or more rows of a database table in the allocated buffer (Chen et al col. 5 lines 34-37, indicating that portions of database tables are stored in the buffer);**
>
> **communicating a message to one or more destination modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Chen et al col. 6 lines 48-51, sending row data contained in a buffer to a node);**
>
> **otherwise:**
>
> **executing a many-rows method to redistribute the one or more rows (Shatdal et al p. 106 col. 1 lines 12-16, the allocated memory is not large enough to fit the table, tuples (i.e. rows) are partitioned into multiple buckets. This is a many-rows method of redistributing, since it is executed when there the number of rows is too large to fit into allocated memory.).**

Applicant respectfully disagrees. Shatdal does not teach or suggest executing a many-rows method to redistribute the one or more rows, as required by claim 11. A many-rows redistribution method is not simply a method that is executed when the number of rows is too large to fit into allocated memory, as suggested in the Office Action. A description of an example of a many-rows method to distribute the one or more rows is in Fig. 5 and accompanying text of the specification. The portion of Shatdal cited in the Office Action does

not teach or suggest such a method. It is admitted in the Office Action that Chen does not disclose a many-rows method to distribute the one or more rows.

Further, claim 11 depends from claim 10 and is patentable for at least the reasons described above for claim 10. Applicant respectfully requests that this rejection be withdrawn.

> **As to claim 17, Chen et al, as modified by Shatdal et al, teaches determining the number of transmitting modules on which the program was invoked (Chen et al col. 6 lines 59-60, determining the number of transmitting modules is inherent, since-each module is notified that reception of the information is complete.).**

Applicant respectfully disagrees. Chen at col. 6, lines 59-60 states that "each node is notified that the reception of information is complete." It is not inherent that the number of transmitting modules on which the program was invoked, as required by claim 17, will be determined inherently in light of this statement. "'Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes, the claim limitations, it anticipates.'" *Perricone v. Medicis Pharmaceutical Corp.*, 432 F.3d 1368, 1376 (Fed. Cir. 2006)(quoting *In re Cruciferous Sprout Litig.*, 301 F.3d 1343, 1349 (Fed. Cir. 2002). It is not necessarily true that simply because each of Chen's nodes is notified that the reception of information is complete, the number of transmitting modules on which the program was invoked will be determined. As such, claim 17 is patentable over Chen. Further, claim 17 depends from claim 10 and is patentable for the reasons set out above for claim 10. Applicant respectfully requests that this rejection be withdrawn.

> **As to claim 18, Chen et al teaches a computer-readable medium containing computer-executable code (col. 4 lines 46-49) for instructing a computer to:**
>
> **allocate a buffer associated with a transmitting processing module, the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables (col. 2 lines 47-50, allocating a communication buffer for transferring rows between nodes, enabling redistribution of rows);**

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows:

store one or more rows associated with a database table in the allocated buffer (col. 5 lines 34-37, indicating that portions of database tables are stored in the buffer);

communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (col. 6 lines 48-51, sending row data contained in a buffer to a node);

Chen et al does not explicitly teach compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows;

execute a many-rows method to redistribute the one or more row.

Shatdal et al teaches compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows (p. 106 col. 1 lines 12-16, determining if a table will fit in allocated memory);

execute a many-rows method to redistribute the one or more rows (p. 106 col. 1 lines 12-16, the allocated memory is not large enough to fit the table, tuples (i.e. rows) are partitioned into multiple buckets. This is a many-rows method of redistributing, since it is executed when there the number of rows is too large to fit into allocated memory.).

Therefore, it would have been obvious to one skilled in the art at the time the invention was made to have modified the method of redistributing rows in a database taught by Chen et al by the step of comparing the size of a buffer to the size of a table of rows to be stored in the buffer, because comparing the size of a buffer to the size of a table of rows to be stored in the buffer while redistributing rows in a database enables dynamic adaptation to grouping selectivities at query evaluation time (Shatdal et al p. 104 col. 1 lines 7-13).

Applicant respectfully disagrees. Chen does not teach or suggest computer-executable code that will instruct a computer to store one or more rows of the allocated buffer if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows, as required by claim 18. Instead, according to the portion of Chen cited in the Office Action, Chen appears to always (i.e., not conditioned on the size of the buffer or the portion of the buffer to be occupied) store portions of tables prior to transmission.

Further, Shatdal does not teach or suggest executing a many-rows method to redistribute the one or more rows, as required by claim 18. A many-rows redistribution method is not simply a method that is executed when the number of rows is too large to fit into allocated memory, as

suggested in the Office Action. A description of an example of a many-rows method to distribute the one or more rows is in Fig. 5 and accompanying text of the specification. The portion of Shatdal cited in the Office Action does not teach or suggest such a method. It is admitted in the Office Action that Chen does not disclose a many-rows method to distribute the one or more rows.

Thus, the combination of Chen and Shatdal suggested in the Office Action is missing at least two elements required by claim 18. Consequently, the Office Action fails to make a prima facie case of obviousness. Applicant respectfully requests that this rejection be withdrawn.

As to claim 26, Chen et al teaches a database management system (Chen et al Figure 2, col. 3, line 10), comprising:

a massively parallel processing system (Figure 2, col. 3, line 10) comprising:

one or more nodes (items 61, 63, 65, and 67, Figure 2);

a plurality of processors, each of the one or more nodes providing access to on or more processors (item 22 in Figure 2, showing a processor in a node); and

a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes (col. 4 lines 42-45);

a set of one or more database tables containing information organized by geographic location (items 62, 66, and 64 in Figure 3, col. 5 lines 5-6); and one or more of the plurality of virtual processes operable to:

allocate a buffer associated with a transmitting processing module, the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables (col. 2 lines 47-50, allocating a communication buffer for transferring rows between nodes, enabling redistribution of rows);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows:

store one or more rows associated with a database table in the allocated buffer (col. 5 lines 34-37, indicating that portions of database tables are stored in the buffer);

communicate a message to one or more destination processing modules, the message comprising at least some of the one or more

rows stored in the allocated buffer (col. 6 lines 48-51, sending row data contained in a buffer to a node);

Chen et al does not explicitly teach compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows;

execute a many-rows method to redistribute the one or more rows.

Shatdal et al teaches compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows (p. 106 col. 1 lines 12-16, determining if a table will fit in allocated memory);

execute a many-rows method to redistribute the one or more rows (p. 106 col. 1 lines 12-16, the allocated memory is not large enough to fit the table, tuples (i.e. rows) are partitioned into multiple buckets. This is a many-rows method of redistributing, since it is executed when there the number of rows is too large to fit into allocated memory.).

Therefore, it would have been obvious to one skilled in the art at the time the invention was made to have modified the method of redistributing rows in a database taught by Chen et al by the step of comparing the size of a buffer to the size of a table of rows to be stored in the buffer, because comparing the size of a buffer to the size of a table of rows to be stored in the buffer while redistributing rows in a database enables dynamic adaptation to grouping selectivities at query evaluation time (Shatdal et al p. 104 col. 1 lines 7-13):

Chen at col. 4, lines 41-45 discloses storing code in RAM on at least one of the computers in the network, which is not the same thing as a plurality of virtual processes, as required by claim 26.

Chen does not teach or suggest storing one or more rows of the allocated buffer if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows, as required by claim 26. Instead, according to the portion of Chen cited in the Office Action, Chen appears to always (i.e., not conditioned on the size of the buffer or the portion of the buffer to be occupied) store portions of tables prior to transmission.

Further, Shatdal does not teach or suggest executing a many-rows method to redistribute the one or more rows, as required by claim 26. A many-rows redistribution method is not simply a method that is executed when the number of rows is too large to fit into allocated memory, as

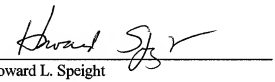
suggested in the Office Action. A description of an example of a many-rows method to distribute the one or more rows is in Fig. 5 and accompanying text of the specification. The portion of Shatdal cited in the Office Action does not teach or suggest such a method. It is admitted in the Office Action that Chen does not disclose a many-rows method to distribute the one or more rows.

Thus, the combination of Chen and Shatdal suggested in the Office Action is missing at least three elements required by claim 26. Consequently, the Office Action fails to make a prima facie case of obviousness. Applicant respectfully requests that this rejection be withdrawn.

SUMMARY

Applicants contend that the claims are in condition for allowance, which action is requested. Applicants do not believe any fees are necessary with the submitting of this response. Should any fees be required, Applicants request that the fees be debited from deposit account number 14-0225, Order Number 069092.0184.

Respectfully submitted,



Howard L. Speight
Reg. No. 37,733
Baker Botts L.L.P.
910 Louisiana
Houston, Texas 77002
Telephone: (713) 229-2057
Facsimile: (713) 229-2757
ATTORNEY FOR APPLICANTS

Date: August 31, 2006